

30 marts

Løsning af \mathcal{NP} -hårde problemer

- Løs til optimalitet i eksponentiel tid
- Find tilnærmet løsning i polynomiel tid

Oversigt

- Optimeringsproblemer \leftrightarrow afgørlighedsproblemer
- Klassen \mathcal{EXP} , og brute-force metoden
- Divide-and-conquer
- Grænseværdier
- Branch-and-bound
- Et konkret eksempel: knapsack problemet

Selv om alle \mathcal{NP} -fuldstændige problemer kan reduceres til hinanden skal specifik struktur udnyttes

1

Afgørlighedsproblem versus optimeringsproblem

Traveling Salesman, afgørlighedsproblem

TSP = $\{ \langle G, c, k \rangle : G = (V, E) \text{ er en komplet graf, } c \text{ er en afstands matrix, } k \text{ er et heltal, } G \text{ har Hamilton-kreds af længde } \leq k \}$

Traveling Salesman, optimeringsproblem

TSP-OPT = $\{ \langle G, c \rangle : G = (V, E) \text{ er en komplet graf, } c \text{ er en afstands matrix, find korteste Hamilton-kreds} \}$

Optimeringsproblemer mere naturlige, "nemmere" at løse

\mathcal{NP} -hård

- Optimeringsproblem er \mathcal{NP} -hårdt \Leftrightarrow
- Tilhørende afgørlighedsproblem er \mathcal{NP} -fuldstændigt

2

Optimeringsproblemer og afgørlighedsproblemer

Optimeringsproblem i *maksimeringsform*

$$z = \max \{ f(x) \mid x \in S \}$$

- $f(x)$ er objektfunktionen
- S er løsningsrummet
- z er optimale løsningsværdi
- optimale løsning, dvs. det x^* hvor $z = f(x^*)$
- verifikation af x^* lige så svært som at løse problem

Optimeringsproblem i *minimeringsform*

$$z = \min \{ f(x) \mid x \in S \}$$

omformes ved at maksimere $-f(x)$

3

Knapsack problem

- *profit* af genstand $p_j \in \mathbb{N}$
- *vægt* af genstand $w_j \in \mathbb{N}$
- *kapacitet* af rygsæk $c \in \mathbb{N}$

$$z = \max \left\{ \sum_{j=1}^n p_j x_j \mid \sum_{j=1}^n w_j x_j \leq c, x_j \in \{0, 1\} \right\}$$

- objektfunktion $f(x) = \sum_{j=1}^n p_j x_j$
- løsningsrum $S = \{ x \in \{0, 1\} \mid \sum_{j=1}^n w_j x_j \leq c \}$

Eksempel

$c = 9, n = 7$

j	1	2	3	4	5	6	7
p_j	6	5	8	9	6	7	3
w_j	2	3	6	7	5	9	4

Optimale løsning: vælg genstande 1 og 4, $z = 15$.

4

Søgebaserede algoritmer

- polynomielle problemer: konstruktiv algoritme
- \mathcal{NP} -hårde problemer: søgebaseret algoritme

Branch-and-bound

- Søgebaseret
- Paradigme
- Total gennemøgning af løsningsrummet (brute-force)
- Matematiske overvejelser udelukker dele af løsningsrummet (grænseværditest)
- Opdeler løsningsrum (divide-and-conquer)
- "Samler" løsninger fra delproblemer

Brute-force metoder

Definition 1 Klassen \mathcal{EXP} : afgørlighedsproblemer som kan løses i eksponentiel tid på deterministisk TM

$$L \in \mathcal{EXP} \Leftrightarrow \exists p(n) : \forall x, |x| = n, x \text{ kan afgøres i tid } 2^{p(n)}$$

For ethvert \mathcal{NP} -problem

- findes et "kort" (dvs. polynomielt) certifikat
- kan et certifikat verificeres i polynomielt tid

Sætning 1 Hvis $L \in \mathcal{NP}$ så gælder også at $L \in \mathcal{EXP}$

Algoritme:

- Gennemløbe alle kombinationer af certifikatet
- For hvert certifikat test om verifikationsalgoritmen returnerer "ja"

Køretid:

- Verifikationsalgoritme $A(x, y)$ hvor $|y| \leq p_2(|x|)$, køretid $p_1(|x|)$
- Alle kombinationer af certifikat $2^{|y|} \leq 2^{p_2(|x|)}$
- Hvert certifikat verificeres i $p_1(|x|)$
- Tid: $2^{p_2(|x|)} \cdot p_1(|x|) = 2^{p_2(|x|)} \cdot 2^{\log p_1(|x|)} = 2^{p_2(|x|) + \log p_1(|x|)}$

Løsningsrummet for et \mathcal{NP} -hårdt problem er begrænset!

Divide and Conquer

- Brute-force metoden for afgørlighedsproblemer
- Divide and conquer for optimeringsproblemer

$$z = \max\{f(x) \mid x \in S\}$$

Sætning 2

- $S = S_1 \cup \dots \cup S_k$ opdeling af S i mindre mængder
- $z_i = \max\{f(x) \mid x \in S_i\}$ løsningsværdi for S_i
- så gælder

$$z = \max_{i=1, \dots, k} z_i$$

Bevis:

$$\begin{aligned} z &= \max\{f(x) \mid x \in S\} \\ &= \max\{f(x) \mid x \in S_1 \cup \dots \cup S_k\} \\ &= \max_{i=1, \dots, k} \max\{f(x) \mid x \in S_i\} \\ &= \max_{i=1, \dots, k} z_i \end{aligned}$$

□

Delproblem: optimeringsproblem begrænset til S_i

Divide and Conquer, eksempel

Knapsack problem, $n = 3$ genstande, kapacitet $c = 6$

j	1	2	3
p_j	6	8	3
w_j	2	6	4

- Løsningsrum

$$S = \{(x_1, \dots, x_n) \mid \sum_{j=1}^n w_j x_j \leq c, x_j \in \{0, 1\}\}$$

- Opdeling af S : x_1 sættes til 0 eller 1

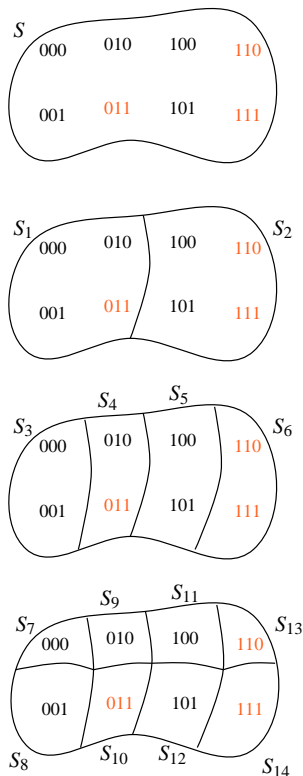
$$\begin{aligned} S_1 &= \{(x_1, \dots, x_n) \subseteq S \mid x_1 = 0\} \\ S_2 &= \{(x_1, \dots, x_n) \subseteq S \mid x_1 = 1\} \end{aligned}$$

- Opdeling af S_1 : x_2 sættes til 0 eller 1

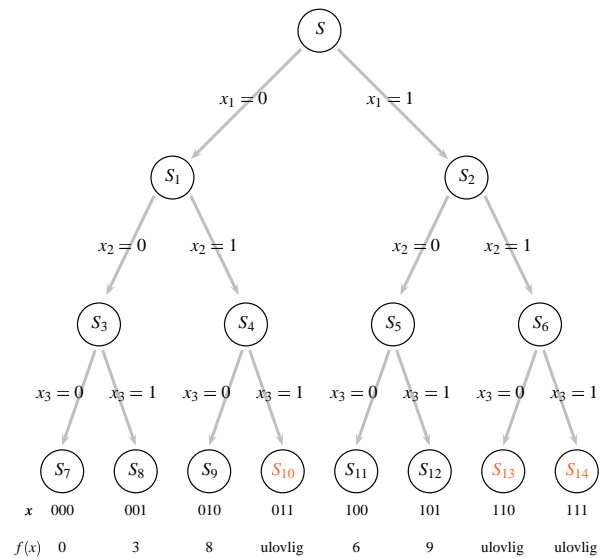
$$\begin{aligned} S_3 &= \{(x_1, \dots, x_n) \subseteq S_1 \mid x_1 = 0, x_2 = 0\} \\ S_4 &= \{(x_1, \dots, x_n) \subseteq S_1 \mid x_1 = 0, x_2 = 1\} \end{aligned}$$

- Illustrere opdelingen af S med et søgetræ
- Hver knude svarer til et delproblem S_i

Optimal løsning til f.x. S_5 : $\max\{z_{11}, z_{12}\} = \max\{6, 9\} = 9$



Figur 1: Opdeling af løsningsrum. De lyse tal svarer til ulovlige løsninger, dvs. løsningsvektorer x hvor $\sum_{j=1}^n w_j x_j > c$.



Figur 2: Søgetræ for knapsack problem. De lyse tal svarer til ulovlige løsninger, dvs. løsningsvektorer x hvor $\sum_{j=1}^n w_j x_j > c$.

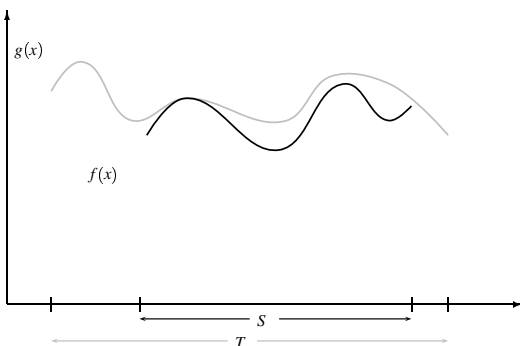
10

Grænseværdier

$$z = \max\{f(x) \mid x \in S\} \quad \text{f.x. } S \text{ delproblem}$$

- nedre grænseværdi $L \in \mathbb{R}: L \leq z$
- øvre grænseværdi $U \in \mathbb{R}: U \geq z$

For enhver løsning $x' \in S$ er $f(x')$ nedre grænseværdi



$$\text{Definition 2} \quad P: z = \max\{f(x) \mid x \in S\}$$

$$R: z_R = \max\{g(x) \mid x \in T\}$$

R er en relaxering af P hvis

- (i) $S \subseteq T$
- (ii) $g(x) \geq f(x)$ for alle $x \in S$

11

Sætning 3 Hvis R er en relaxering af P så gælder $z_R \geq z$.

Bevis

- x^* opt. løsning for $P: f(x^*) = z$
- Fra (ii) da $x^* \in S$ gælder $g(x^*) \geq f(x^*)$
- Fra (i) gælder $x^* \in T$
- $z_R = \max\{g(x) \mid x \in T\} \geq g(x^*) \geq f(x^*) = z$

Grænseværdier

- Løsning z_R til relaxering R er øvre grænseværdi U
- Ofte kræves at relaxering kan løses effektivt

Grænseværditest

Lad U være øvre grænseværdi for S_i
Hvis $U \leq z$ så findes ikke forbedrende løsning i S_i

12

Divide and conquer

$$S = S_1 \cup \dots \cup S_k \text{ opdeling af } S$$
$$z_i = \max\{f(x) \mid x \in S_i\}$$

Sætning 4

- \mathcal{L}_i er nedre grænseværdi for delmængde S_i
- da er $\mathcal{L} = \max_{i=1, \dots, k} \mathcal{L}_i$ en nedre grænseværdi for S

Bevis Da $\mathcal{L}_i \leq z_i$ har vi $\mathcal{L} = \max_{i=1, \dots, k} \mathcal{L}_i \leq \max_{i=1, \dots, k} z_i = z$

Sætning 5

- \mathcal{U}_i er øvre grænseværdi for delmængde S_i
- da er $\mathcal{U} = \max_{i=1, \dots, k} \mathcal{U}_i$ en øvre grænseværdi for S

Bevis Da $z_i \leq \mathcal{U}_i$ har vi $\mathcal{U} = \max_{i=1, \dots, k} \mathcal{U}_i \geq \max_{i=1, \dots, k} z_i = z$

Sætning 6 $f(x)$ heltallig for ethvert $x \in S$

- \mathcal{U} er en øvre grænseværdi
- da er også $\lfloor \mathcal{U} \rfloor$ en øvre grænseværdi

13

Knapsack problem, øvre grænseværdi

Fraktionelle knapsack problem i Cormen

$$\mathcal{U}_{\text{kp}}^1 = \max \left\{ \sum_{j=1}^n p_j x_j \mid \sum_{j=1}^n w_j x_j \leq c, 0 \leq x_j \leq 1 \right\}$$

Originale knapsack problem

$$f(x) = \sum_{j=1}^n p_j x_j$$

$$S = \left\{ (x_1, \dots, x_n) \mid \sum_{j=1}^n w_j x_j \leq c, x_j \in \{0, 1\} \right\}$$

Fraktionelle knapsack problem

$$g(x) = \sum_{j=1}^n p_j x_j$$

$$T = \left\{ (x_1, \dots, x_n) \mid \sum_{j=1}^n w_j x_j \leq c, 0 \leq x_j \leq 1 \right\}$$

Da $f(x) = g(x)$ er kriterium (ii) i definition 2 overholdt. Endvidere er $S \subseteq T$, hvorfor kriterium (i) også er opfyldt.

14

Branch-and-bound

$$z = \max\{f(x) \mid x \in S\}$$

```
1   $\mathcal{L} := -\infty; L := \{S\}$ 
2  while  $L \neq \emptyset$ 
3    vælg et delproblem  $S_i$  fra  $L$ 
4     $L := L \setminus \{S_i\}$ 
5    if  $S_i \neq \emptyset$  then
6      find en øvre grænseværdi  $\mathcal{U}(S_i)$ 
7      find en lovlig løsning  $x$  i  $S_i$ 
8      if  $\mathcal{U}(S_i) > \mathcal{L}$  then
9        if  $f(x) > \mathcal{L}$  then  $\mathcal{L} := f(x); x^* := x$ 
10       opdel  $S_i$  i delproblemer  $S_i^1, \dots, S_i^k$ 
11       tilføj delproblemerne til  $L$ , dvs. sæt
            $L := L \cup \{S_i^1, \dots, S_i^k\}$ 
12     endif
13   endif
14 endwhile
```

- L en liste af delproblemer S_i givet ved de tilhørende løsningsrum. Listen L vil ofte være organiseret som en prioritetskø.
- Vi kalder delproblemerne i L for *åbne delproblemer*, mens delproblemer der allerede er blevet behandlet kaldes *lukkede delproblemer*.
- global nedre grænseværdi \mathcal{L} , tilhørende løsning x^* .

15

Branch-and-bound

En branch-and-bound algoritme for et maksimeringsproblem vil derfor bestå af følgende fire komponenter:

1. En *øvre grænseværdifunktion* som for et givet delproblem returnerer en øvre grænse på værdien af den bedste løsning vi kan finde i delrummet.
2. En *nedre grænseværdi* som på ethvert tidspunkt angiver den hidtil bedst kendte løsning.
3. En *søgestrategi* som fastlægger en rækkefølge for behandlingen af delproblemer.
4. En *forgreningsregel* som for alle delproblemer som ikke kan forkastes af grænseværditesten, angiver hvorledes det tilhørende løsningsrum skal opdeles. Dermed skabes to eller flere nye delproblemer.

16